

**REMARKS****Specification Objections**

The Examiner has objected to the specification because of the following informalities:

- 1) “Sun Microsystems”, cited in [0007], Line 1 is a registered trademark; and
- 2) “The node 305 passes the software version information”, cited in [0080], Line 3, should be corrected as “The node 302 passes the software version information”

Applicant cannot find either of these references in Applicant’s specification. In fact, the term “Sun Microsystems” is never utilized in Applicant’s specification. Furthermore, paragraph [0007] in Applicant’s specification, consists only of the simple introductory line for the brief description of the drawings.

Additionally, there is no “node 305” or “node 302” in Applicant’s specification. Actually, there are no nodes, period, in the entire specification (i.e. no use of the term “node”). More specifically, Applicant’s specification does not even reach paragraph [0080] (Applicant’s specification is 37 paragraphs long, as shown on the USPTO website).

Thus, Applicant assumes these two objections to the specification as set forth by the Examiner were made in error and refer to another application. Therefore, Applicant respectfully requests the Examiner remove both objections to the specification.

**Claim Rejections – 35 U.S.C. § 102**

The Examiner has rejected claims 1-2, 14-15, 23, and 31 under 35 U.S.C. §102(e), as being anticipated by Roediger et. al. (US Patent 6,938,249 B2) (“Roediger”). For the reasons set forth below, Applicant asserts that the cited reference fails to anticipate Applicant’s invention as claimed in claims 1-2, 14-15, 23, and 31.

The title of the Roediger patent is “Compiler Apparatus and Method for Optimizing Loops in a Computer Program.” (Emphasis added) The fact that Roediger teaches a compiler method and apparatus is important. Specifically, Roediger teaches a compiler that includes a “profile-based loop optimizer [that] generates an execution frequency table for each loop that gives more detailed profile data that allows making a more intelligent decision regarding if and how to optimize each loop in the computer program.” (Roediger abstract)

With respect to independent claim 1 in the presently claimed invention, Applicant teaches and claims:

“A method, comprising collecting a loop trip count continuously during runtime of a region of code being executed that contains a loop, categorizing the trip count to identify one or more code modification techniques applicable to the loop, and dynamically applying, during the same runtime, the one or more applicable code modification techniques to alter the code that relates to the loop.”

(Claim 1) (Emphasis added)

Applicant asserts that Roediger does not anticipate Applicant’s invention as claimed in independent claim 1 because Roediger does not teach at least *collecting a loop*

*trip count continuously during runtime and dynamically applying, during the same runtime, the one or more applicable code modification techniques.*

Applicant's method teaches dynamically collecting information and applying the code optimization techniques during the same runtime of the loop region of code. This is because Applicant's method relates to a dynamic binary translator (this terminology is utilized in the title of Applicant's presently claimed invention).

The method in Roediger, on the other hand, applies any loop optimizations to a compiler. The operation of a compiler is inherently different and inferior to a dynamic binary translator regarding real-time optimizations because it is impossible for a compiler to dynamically apply code optimizations during runtime of the code. Rather, with a compiler, such as the compiler in Roediger, a profiler must instrument the code, the user then provides sample inputs for the code and gathers profile data, and the compiler recompiles the code with potential optimizations applying the profile data to enhance the code. This method is shown in Roediger, Figure 2.

The method in Roediger does not *collect a loop trip count continuously during runtime and dynamically apply, during the same runtime, the one or more applicable code modification techniques*. This is because this method is impossible for a compiler to accomplish. Thus, the method in Roediger requires many additional steps that are not necessary in Applicant's invention as claimed in independent claim 1. Namely, the method in Roediger requires:

- Instrumenting program code and running sample data to gather compiler profile data (first time code is run through, albeit only with sample data and not actually during real world code usage).

- Compiling the code using gathered profile data to implement any potential optimizations. (compiling requires a period of time separate from runtime – i.e. code being compiled is not being run while compiled).
- Running recompiled code (second time code is run through, thus there is no continuous single runtime).

The differences between a compiler (Roediger) and a dynamic binary translator (Applicant's presently claimed invention) are quite fundamental. The most significant difference is that a dynamic binary translator is specifically meant to modify and translate code all during runtime and a compiler is limited to compiling code not during runtime. Due to these differences and because of the fact that the method in Roediger cannot at least *collect a loop trip count continuously during runtime and dynamically apply, during the same runtime, the one or more applicable code modification techniques* (since, as described above, this is impossible for a compiler), Applicant respectfully submits that Roediger does not anticipate Applicant's invention as claimed in independent claim 1.

In regard to independent claims 15, 23, and 31, Applicant respectfully submits that Roediger does not anticipate Applicant's invention at least for the same reasons as independent claim 1. Again, Roediger does not teach at least *collecting a loop trip count continuously during runtime and dynamically applying, during the same runtime, the one or more applicable code modification techniques*. Thus, Applicant respectfully submits that Roediger does not anticipate independent claims 15, 23, and 31 of the presently claimed invention.

Claims 2 and 14 depend from and further limit independent claim 1. Thus, for at least the same reasons advanced above with respect to independent claim 1, Applicant respectfully submits that Roediger does not anticipate claims 2 and 14.

Thus, Applicant respectfully requests withdrawal of the 35 U.S.C. 102(e) rejection of claims 1-2, 14-15, 23, and 31.

### **Claim Rejections – 35 U.S.C. § 103**

The Examiner has rejected claims 3, 5-8, 12-13, 16-22, 24-25, 27-30, 32-33, and 35-38 under 35 U.S.C. § 103(a), as being unpatentable over Roediger in view of Chen et. al. (Dynamic Trace Selection Using Performance Monitoring Hardware Sampling, March 2003, IEEE) (“Chen”). For the reasons set forth below, Applicant asserts that the cited references fail to teach, suggest, or render obvious Applicant’s invention as claimed in claims 3, 5-8, 12-13, 16-22, 24-25, 27-30, 32-33, and 35-38.

Claims 3, 5-8, 12-13, 16-22, 24-25, 27-30, 32-33, and 35-38 depend from and further limit independent claims 1, 15, 23, and 31, respectively. Thus, for at least the same reasons advanced above with respect to independent claims 1, 15, 23, and 31, Applicant respectfully submits that Roediger and Chen, each taken alone or in combination do not teach, suggest, or render obvious claims 3, 5-8, 12-13, 16-22, 24-25, 27-30, 32-33, and 35-38.

Thus, Applicant respectfully requests withdrawal of the 35 U.S.C. 103(a) rejection of claims 3, 5-8, 12-13, 16-22, 24-25, 27-30, 32-33, and 35-38.

The Examiner has rejected claims 4, 9-11, 26, and 34 under 35 U.S.C. § 103(a), as being unpatentable over Roediger in view of Chen and further in view of Ghosh et. al.

(Integrating High-Level Optimization in a Production Compiler: Design and Implementation Experience, April 2003, Springer-Verlag Berlin Heidelberg) (“Ghosh”). For the reasons set forth below, Applicant asserts that the cited references fail to teach, suggest, or render obvious Applicant’s invention as claimed in claims 4, 9-11, 26, and 34.

Claims 4, 9-11, 26, and 34 depend from and further limit independent claims 1, 23, and 31, respectively. Thus, for at least the same reasons advanced above with respect to independent claims 1, 23, and 31, Applicant respectfully submits that Roediger, Chen, and Ghosh, each taken alone or in combination do not teach, suggest, or render obvious claims 4, 9-11, 26, and 34.

Thus, Applicant respectfully requests withdrawal of the 35 U.S.C. 103(a) rejection of claims 4, 9-11, 26, and 34.

**CONCLUSION**

Applicant respectfully submits that all rejections have been overcome and that all pending claims are in condition for allowance.

If there are any additional charges, please charge them to our Deposit Account Number 50-0221. If a telephone conference would facilitate the prosecution of this application, the Examiner is invited to contact Derek J. Reynolds at (916) 356-5374.

Respectfully Submitted,

Date: September 26, 2007 /Derek J. Reynolds/  
Derek J. Reynolds  
Reg. No.: 60,104

Intel Corporation  
c/o Intellevate, LLC  
P.O. Box 52050  
Minneapolis, MN 55402  
(916)356-5358